

UMA FERRAMENTA COMPUTACIONAL PARA O AGENDAMENTO DE OPERAÇÕES DO PROGRAMA DE ACESSIBILIDADE ESPECIAL PORTA A PORTA - PRAE

Saulo de Tarso Alves Dantas^{a*}, Caroline Rocha^a, Daniel Aloise^a,

José Claudio Galvão^b, Ana Maria da Silva Martins^c

^aUniversidade Federal do Rio Grande do Norte – UFRN, Natal – RN

^bSecretaria Municipal da Saúde – SMS-PRAE, Natal – RN

^cPronto Socorro Clóvis Sarinho – PSCS, Natal – RN

Resumo

Em todo o mundo, a demanda por serviços de transporte para pessoas portadoras de necessidades especiais, idosos, e pessoas com mobilidade reduzida vêm crescendo nos últimos anos. A população está envelhecendo, os governos precisam se adaptar a esta realidade, e este fato pode significar oportunidade de negócios para as companhias. Dentro deste contexto está inserido o Programa de Acessibilidade Especial porta a porta – PRAE do município de Natal-RN. A pesquisa presente neste trabalho procura desenvolver um modelo de programação capaz de auxiliar o processo de tomada de decisão dos gestores deste serviço de transporte. Para tanto, foi criado um algoritmo baseado em métodos de geração de soluções aproximativas conhecidas como heurísticas. O objetivo do modelo é incrementar o número de pessoas atendidas pelo PRAE, dada a frota disponível, gerando programações de roteiros otimizadas. O PRAE consiste em um problema de roteirização e programação de veículos do tipo dial-a-ride – DARP, um dos tipos mais complexos dentre os problemas de roteirização. A validação do método de resolução foi feita mediante a comparação entre os resultados auferidos pelo modelo computacional e a programação manual real atual. Os resultados mostraram que o modelo idealizado neste trabalho foi capaz de elevar a capacidade de atendimento deste serviço de transporte.

Palavras-Chaves: PRAE. Mobilidade reduzida. Modelo computacional. Heurística.

DARP. Dial-a-ride.

Abstract

Worldwide, the demand for transportation services for persons with disabilities, the elderly, and persons with reduced mobility have increased in recent years. The world population is aging, governments need to adapt to this reality, and this fact could mean business opportunities for companies. Within this context is inserted the Programa de Acessibilidade Especial porta a porta – PRAE, a door to door public transportation service from the city of Natal-RN in Brazil. The research presented in this paper seeks to develop a computational model which can assist the process of decision making of managers of the shuttle. To that end, it was created an algorithm based on methods of generating approximate solutions known as heuristics. The purpose of the model is to increase the number of people served by the PRAE, given the available fleet, generating optimized schedules routes. The PRAE is a problem of vehicle routing and scheduling of dial-a-ride - DARP, on of the most complex type among the routing problems. The validation of the method of resolution was made by comparing the results derived by the computational model and the currently programming manual method. The results shown that the model created in this paper was able to increase the current capacity of the transport service.

Key Words: PRAE. Reduced mobility. Computacional Model. Heuristic. DARP. Dial-a-ride.

*Autor para correspondência: e-mail: saulo.alvesdantas@yahoo.com.br

1. Introdução

A sociedade moderna vem se preocupando mais com assuntos concernentes com a igualdade de oportunidades e o bem-estar geral. Dentro deste contexto, destaca-se uma fração da população que devido a alguma restrição física de mobilidade acaba sofrendo com barreiras ambientais e atitudinais que bloqueiam sua plena e efetiva participação na sociedade em igualdade de oportunidades com as demais pessoas. Esta parcela da sociedade é geralmente representada pelas pessoas que possuem algumas funções motoras comprometidas ou reduzidas, mesmo que temporariamente, por indivíduos que precisem de assistência para se locomover devido a inacurácia visual, entre outros casos onde a mobilidade fica restrita. No último levantamento feito pelo Instituto Brasileiro de Geografia e Estatística – IBGE, aproximadamente 6,5% da população brasileira possuía algum tipo de deficiência visual, auditiva, motora e mental (CENSO, 2012).

Um dos problemas que o indivíduo com mobilidade reduzida precisa enfrentar todos os dias é o de acesso rápido ao transporte público adaptado à sua condição. Não é difícil encontrar uma situação de frustração em que um portador de necessidades especiais precisa esperar mais de uma hora por um transporte adaptado. É notável que o indivíduo com mobilidade reduzida ainda possui desvantagens quando comparado a um indivíduo que não precisa se programar com tamanha antecedência para conseguir cumprir algum horário agendado, ou mesmo chegar ao trabalho. Outro ponto frequentemente destacado pelos usuários do transporte público são os obstáculos que se apresentam no trajeto de suas residências ao ponto de ônibus. As calçadas irregulares e sem manutenção, a falta de rampas, e calçadas reformadas sem uniformidade de altura pelos próprios moradores são alguns dos fatores que dificultam o acesso. Esta realidade está mudando gradativamente, e cada vez mais atenção é direcionada para esta fração da sociedade.

Uma amostra da preocupação com o tema é a própria definição de “pessoa com mobilidade reduzida” que está prevista na Lei nº 10.098/00, com normas gerais e critérios básicos para assegurar a acessibilidade das pessoas portadoras de deficiência ou com mobilidade reduzida, como sendo aquela a que temporariamente tem limitada sua capacidade de se relacionar com o meio e de utilizá-lo (art. 2º, III). Ainda segundo Cordeau e Laporte (2003a), o crescimento da procura por estes serviços pode ser motivado por dois fatores. A primeira razão é o crescimento da população de idosos no mundo, e a segunda é a preocupação das entidades governamentais com a criação de serviços de apoio a saúde doméstica e assistência às pessoas.

Apesar da crescente preocupação global relativo ao presente tema, os serviços de transporte especial ainda se encontram muito ineficientes como mostram Melachrinoudis *et al.* (2007). Os custos de transporte ainda são elevados devido a viagens inoportunas, longos tempos de espera, viagens não programadas, ociosidade e subutilização da capacidade dos veículos. Isso repele o interesse por investidores em um serviço que pode ser lucrativo.

É dentro desta realidade que se insere um dos poucos programas de transporte a pessoas com algum tipo de mobilidade reduzida que existem no Brasil, o Programa de Acessibilidade Especial porta a porta – PRAE, do município de Natal, capital do estado do Rio Grande do Norte. Suas atividades se iniciaram no ano de 2008 envolvendo empresas concessionárias que firmaram o compromisso de disponibilizar veículos adaptados para pessoas com deficiências sem cobrança de tarifas (PRAE, 2013).

A configuração do serviço de transporte do PRAE é classificada na literatura como um problema de roteirização e programação de veículos do tipo *dial-a-ride* – DARP (CORDEAU; LAPORTE, 2007). O DARP é um problema de otimização combinatória (POC), onde o objetivo consiste em encontrar valores para um conjunto de variáveis de decisão relacionadas por uma função objetivo que precisa ser otimizada (minimizada ou maximizada) atendendo a um conjunto de restrições. Para resolver um POC em um tempo apreciável é preciso se utilizar de técnicas computacionais mais sofisticadas do que simplesmente listar todas as soluções possíveis e procurar, dentre elas, a melhor. O DARP é entendido pela literatura como um problema de classe NP-Difícil, o que significa dizer que não se conhecem algoritmos que consigam resolvê-lo em tempo polinomial (BAUGH *et al.*, 1998).

Diversos modelos com algoritmos para soluções exatas já foram desenvolvidos com o passar do tempo, como por exemplo: um *Branch-and-Bound* por Balas e Toth (1985), um *Branch-and-cut* por Padberg e Rinaldi (1991), e um algoritmo de Programação Dinâmica por Saxe (1980). No entanto, a grande quantidade de variáveis encontrada nos problemas reais torna o tempo, despendido na busca da solução ótima, pelos métodos exatos, inviável para o ritmo do processo de tomada de decisão das atividades diárias. Para os casos com grandes quantidades de variáveis, recorre-se ao uso de algoritmos heurísticos que apesar de não garantirem que a melhor solução possível para o problema (solução ótima) seja encontrada, conseguem encontrar boas soluções em um tempo compatível com as necessidades de seu uso.

O objetivo desta pesquisa é desenvolver uma ferramenta computacional baseada em técnicas de otimização para auxiliar no processo de planejamento logístico. O algoritmo desenvolvido procura maximizar o número de pessoas atendidas pelo serviço de transporte

especial para pessoas com mobilidade reduzida se utilizando de métodos heurísticos de resolução em tempo adequado às necessidades dos gestores. Para alcançar este objetivo é preciso que o modelo consiga gerar soluções que aloquem uma quantidade maior de solicitações na programação das rotas do que conseguem fazer os gestores que planejam o agendamento semanal, atendendo a todas as restrições impostas pelo problema. Ademais, a ferramenta logística deve reduzir o tempo despendido pelos gestores na tarefa de planejamento permitindo que haja tempo adicional para ajustes estratégicos e outras atividades.

2. Referencial teórico

O DARP possui diversos modos de ser resolvido e todos eles foram concebidos partindo de técnicas de otimização combinatória desenvolvidas para a solução de um conjunto de problemas classificados na literatura como de roteirização ou roteamento de veículos - PRVs, o nome varia dentre as traduções para o português (BODIN *et al.*, 1983). O PRV descreve de maneira geral uma classe de problemas em que uma determinada frota de veículos precisa alcançar várias localizações espalhadas no espaço físico através de um conjunto de caminhos disponíveis que ligam os pontos por onde os veículos precisam passar. O PRV teve sua apresentação na literatura com o estudo de Dantzig e Ramser (1959) e seu problema de distribuição de combustível com vários pontos de abastecimento e vários pontos de destino.

Um dos desafios da modelagem dos problemas de roteirização ocorre quando o serviço consiste em transportar pessoas portadoras de necessidades especiais, idosos ou indivíduos com mobilidade reduzida. Este serviço exige muita flexibilidade pois geralmente o veículo busca o passageiro na porta de casa e o transporta para o lugar desejado nos horários especificados pelo usuário. O condutor recebe uma lista de pessoas e lugares e precisa decidir a melhor sequência de embarques e desembarques de maneira a não demorar demais para não provocar atrasos, precisa mensurar quantas pessoas pode transportar e não ultrapassa sua carga horária de trabalho. Por sua vez os usuários, devido a condições físicas, não podem dispendir mais que um limite tolerável de tempo dentro do veículo, entre outras muitas restrições que o tipo de serviço possui.

2.1. Principais características do DARP

Os serviços do tipo *dial-a-ride* podem ser divididos em dois tipos: estático ou dinâmico. No caso estático (caso do PRAE), os interessados realizam um cadastro prévio para gerar uma solicitação de transporte permitindo que a demanda dos usuários seja conhecida com

antecedência, enquanto que no caso dinâmico, a demanda pelo serviço é imediata e vai ocorrendo gradualmente ao longo do dia, ou seja, as rotas e a programação dos veículos são construídas em tempo real sofrendo adaptações constantes. Na prática, os problemas dinâmicos puros raramente ocorrem porque parte das solicitações feitas já são previamente conhecidas.

Uma tendência comum nos modelos DARP é permitir que os usuários determinem uma janela de tempo tanto para o momento de embarque nas suas residências quanto para o momento de desembarque no local desejado. Essa limitação é uma maneira de manter um nível de qualidade adequado do serviço de transporte já que a carga transportada é composta somente de pessoas.

Os DARPs ainda podem variar segundo a forma de embarque e desembarque que é executada. Relacionada com a capacidade do transporte, as variações são: *single* ou *mixed loads*. No problema *single load*, o veículo é esvaziado no ponto de destino para só então proceder com a coleta do próximo usuário. No caso *mixed load*, os passageiros são coletados e entregues de forma continuada respeitando a capacidade máxima que o veículo comporta de pessoas sentadas, este último consiste no caso do PRAE.

A maioria dos estudos feitos concernentes com o DARP assume a disponibilidade de uma frota de m veículos idênticos e todos com origem em uma mesma garagem. Segundo Cordeau e Laporte *et al.* (2007), esta hipótese preliminar normalmente reflete a realidade e ajuda a desenvolver os modelos e algoritmos. No entanto, é importante notar que diversas situações podem ocorrer na prática, como é o caso com vários pontos de partida ou garagens (*multi-depot*), assim como vários tipos de veículos onde alguns são feitos para acomodar apenas passageiros em cadeira de rodas, outros possuem apenas uma vaga para acamados e acompanhantes, e ainda os que possuem vagas mistas para diferentes tipos de necessidades, ou seja, veículos heterogêneos. O problema do PRAE é exatamente o do tipo mais complexo dentre as variações existentes do DARP, ele possui uma frota limitada de veículos heterogêneos com capacidades distintas partindo de diversos pontos ou garagens.

2.2. Modelo Clássico do DARP

Geralmente os problemas do tipo *dial-a-ride* possuem múltiplos objetivos onde a construção da função objetivo e das restrições depende de cada caso específico. Por exemplo, Toth e Vigo (1997) consideraram minimizar a distância total percorrida pelo veículo. Por outro lado, Baugh *et al.* (1998) trabalharam para minimizar os custos totais da rota, as violações da janela de tempo (qualidade do serviço), e o número de veículos usados. Já Jørgensen *et al.* (2007) procuraram

minimizar o tempo de viagem dos passageiros e o tempo de espera dos passageiros. Aldaihani e Dessouky (2003) resolveram o DARP minimizando a distância total percorrida e o tempo total de viagem de todos os passageiros. Ademais, cada um desses problemas possuem características próprias quanto a heterogeneidade da frota, a quantidade de garagens, a casos estáticos ou dinâmicos, e a existência de rotas pré-fixadas. Isso mostra que o DARP não é um problema que pode ser resolvido com um modelo genérico, é preciso adequar as formulações às características de cada problema.

A qualidade do serviço é geralmente traduzida pelas seguintes restrições: a duração da rota não pode exceder a jornada do motorista; o tempo de espera para embarque deve atender à tolerância do cliente; atrasos devem ser evitados; o passageiro não pode passar muito tempo dentro do veículo, entre outras mais específicas de cada situação.

O DARP é geralmente modelado em um grafo completo $G = (V, A)$, onde V representa o conjunto de todos os vértices e A o conjunto de todas as arestas. Para cada aresta (i, j) é atribuído um peso c_{ij} . Um total de n solicitações é feita e cada solicitação representa um par de vértices, sendo sempre um de embarque e outro de desembarque $\{i, n + i\}$. O tempo total de viagem da rota não pode ser maior que o tempo máximo da rota estabelecido T . Da mesma forma, o tempo máximo de viagem do usuário L dentro do veículo não pode ser ultrapassado. Os pedidos são atendidos por uma frota de m veículos onde cada veículo k possui uma capacidade máxima Q_k de assentos. Os veículos iniciam a jornada da garagem no ponto denotado por 0 e encerram no mesmo ponto, mas agora denotado por $2n + 1$. Em cada vértice de embarque i existe uma quantidade de usuários com carga $(q_i > 0)$ para ser transportados, sendo q_i a carga ou a quantidade de vagas que o(s) usuário(s) ocupa(m) no veículo. Em contrapartida, para cada vértice de desembarque $(n + i)$ é atribuído uma carga $q_{n+i} = -q_i$ representando a saída do(s) usuário(s) do veículo e a efetivação da(s) solicitação(ões).

Os vértices são associados a uma janela de tempo $[e_i l_i]$, onde e_i é o instante de início da janela de tempo e l_i o do término, podendo variar de acordo com o tipo do pedido, *outbound* ou *inbound*. Na literatura, os pedidos de *outbound* denotam transporte de origem na residência de um passageiro. Já os pedidos de *inbound* representam o inverso, o transporte do passageiro de onde estiver de volta a sua residência. É preciso fazer esta distinção pelo fato de vários métodos de resolução do DARP determinarem diferentes janelas de tempo dependendo do lugar e do tipo do pedido. Por exemplo, janelas de tempo mais estreitas podem ser aplicadas nos pontos que não são a residência do passageiro para evitar atrasos ou espera demasiada. Isso se

traduz, para a construção do algoritmo, em janelas mais restritas no ponto de desembarque em pedido de *outbound* e no ponto de embarque em pedidos de *inbound*.

Quando se faz essa diferenciação, os vértices com a janela de tempo mais estreita é também chamado de **vértice crítico do pedido** (CORDEAU; LAPORTE, 2003b). Define-se por d_i o tempo de serviço decorrido no processo de embarque e desembarque do usuário em cada vértice. Detona-se por D_i o instante de saída do vértice i , e A_j o instante de chegada ao vértice subsequente j , dado por $A_j = D_i + t_{ij}$, onde t_{ij} é o tempo decorrido no deslocamento de i para j . Ao alcançar j , o serviço de embarque só se inicia caso o instante de chegada seja maior ou igual ao início da janela de tempo deste vértice, que agora é representado novamente por i pois se trata do vértice corrente, sendo o instante do início do serviço dado por $B_i = \max\{A_i, e_i\}$. Caso a chegada ocorra antes do início da janela de tempo o veículo espera no vértice corrente um tempo $W_i = B_i - A_i$. O instante da partida do veículo do vértice corrente i corresponde a $D_i = B_i + d_i$ e o tempo em que um passageiro permanece no veículo, ou seja, o tempo de viagem é $L_i = B_{n+i} - D_i$, onde B_{n+1} é o instante do início do serviço no vértice de desembarque $n + i$, e D_i o instante de saída do vértice i da solicitação $(i, n + i)$.

Além disso, é preciso calcular a carga corrente do veículo em cada vértice y_i ou o número de assentos ocupados sempre que o veículo parte de um vértice i para saber se em algum momento da rota a capacidade máxima do veículo é violada.

O cálculo da duração total da rota k é dada pela diferença entre o instante do início do serviço no último vértice da rota B_{2n+1}^k e o instante do início do serviço no primeiro vértice da rota B_0^k , ou seja, $B_{2n+1}^k - B_0^k$, onde 0 e $2n + 1$ representam a saída e a chegada na garagem.

Na literatura é comum encontrar os métodos de resolução para o DARP classificados em métodos exatos e métodos heurísticos.

Um problema de otimização combinatória real é, por vezes, classificado como um problema NP-Difícil devido ao grande número de variáveis. Estes são exatamente os problemas para os quais não existe um algoritmo que seja capaz de resolvê-lo em tempo polinomial. Os métodos de resolução exatos existentes conseguem resolver problemas NP-Difíceis em tempo exponencial. Devido a este fato, o tempo de resolução é ineficiente para uma grande parte dos propósitos práticos. Por isso, o uso de métodos aproximativos para resolver problemas de otimização combinatória tem recebido maior atenção (BLUM; ROLI, 2003).

2.3. Modelo de Referência

Diversos modelos têm sido desenvolvidos na tentativa de resolver o DARP de forma exata como visto na seção anterior. O que será apresentado neste item é a formulação de 3 índices de Cordeau (2006), que é a que mais se aproxima do formato do funcionamento do PRAE. As variáveis das formulações seguem a mesma notação do modelo clássico apresentado na seção 2.2.

A formulação é baseada em um grafo direcionado $G = (V, A)$, ou seja, o caminho de ida de um ponto ao outro não é o mesmo de volta. O conjunto dos vértices V é dividido em $\{0, 2n + 1, P, D\}$ em que 0 e $2n + 1$ representam a garagem, sendo respectivamente um vértice de partida e um vértice de chegada. $P = \{1, \dots, n\}$ é o conjunto dos vértices de embarque e $D = \{n + 1, \dots, 2n\}$ é o conjunto dos vértices de desembarque. Uma solicitação é representada pelo par $(i, n + i)$, onde $i \in P$ e $n + i \in D$. Para cada vértice $v_i \in V$ é associado uma carga q_i , sendo $q_0 = q_{2n+1} = 0$, $q_i \geq 0$ para todo $i = 1, \dots, n$, e $q_i = -q_{i-n}$ para todo $i = n + 1, \dots, 2n$, além disso, a duração do serviço de embarque e desembarque para cada solicitação é $d_i \geq 0$ com $d_0 = d_{2n+1} = 0$. O conjunto das arestas é definido como $A = \{(i, j): i = 0, j \in P, \text{ ou } i, j \in P \cup D, i \neq j \text{ e } i \neq n + j, \text{ ou } i \in D, j = 2n + 1\}$. A capacidade do veículo é Q_k . O tempo máximo em que um passageiro pode permanecer no veículo é denotado por L e a janela de tempo do vértice i é $[e_i, l_i]$. Cada rota é representada por um veículo k . A duração máxima da rota $k \in K$ é definida por T_k . O custo de atravessar a aresta (i, j) com o veículo k é c_{ij}^k . O tempo no qual o veículo inicia o serviço de embarque ou desembarque no vértice i é denotado como u_i^k . A carga do veículo quando partindo do vértice i é w_i^k . O tempo de viagem de cada usuário, ou seja, de cada solicitação i correspondente ao par da solicitação $(i, n + i)$ é r_i^k .

O modelo matemático é o seguinte:

$$\text{Minimizar } \sum_{k \in K} \sum_{i \in V} \sum_{j \in V} c_{ij}^k x_{ij}^k \quad (1)$$

Sob as restrições:

$$\sum_{k \in K} \sum_{j \in V} x_{ij}^k = 1 \quad (i \in P) \quad (2)$$

$$\sum_{j \in V} x_{ij}^k - \sum_{j \in V} x_{n+i,j}^k = 0 \quad (i \in P, k \in K) \quad (3)$$

$$\sum_{i \in V} x_{0i}^k = \sum_{i \in V} x_{i,2n+1}^k = 1 \quad (k \in K,) \quad (4)$$

$$\sum_{j \in V} x_{ji}^k - \sum_{j \in V} x_{ij}^k = 0 \quad (i \in P \cup D, k \in K) \quad (5)$$

$$u_j^k \geq (u_i^k + d_i + t_{ij})x_{ij}^k \quad (i, j \in V, k \in K,) \quad (6)$$

$$w_j^k \geq (w_i^k + q_i)x_{ij}^k \quad (i, j \in V, k \in K) \quad (7)$$

$$r_i^k \geq (u_{n+i}^k - (u_i^k + d_i)) \quad (i \in P, k \in K) \quad (8)$$

$$u_{2n+1}^k - u_0^k \leq T_k \quad (k \in K) \quad (9)$$

$$e_i \leq u_i^k \leq l_i \quad (i \in V, k \in K) \quad (10)$$

$$t_{i,n+i} \leq r_i^k \leq L \quad (i \in P, k \in K) \quad (11)$$

$$\max\{o, q_i\} \leq w_i^k \leq \min\{Q_k, Q_k + q_i\} \quad (i \in V, k \in K) \quad (12)$$

$$x_{ij}^k = 0 \text{ ou } 1 \quad (i, j \in V, k \in K) \quad (13)$$

Na formulação, as restrições 2 e 3 asseguram que cada solicitação é encaixada em apenas uma única rota, ou seja, evita replicação de solicitações já alocadas em mais de uma rota e obriga que o número de arestas chegando em um vértice seja a mesma quantidade de saída. Vale lembrar que o modelo se baseia em um grafo direcional.

As restrições 4 e 5 garantem que cada veículo começa e termina sua rota na garagem, e que o número de embarques de todos os vértices seja igual ao número de desembarques. As restrições de 6, 7 e 8 definem o tempo do início do serviço de embarque e desembarque, a carga no veículo e o tempo de viagem do usuário, enquanto que as restrições de 9, 10, 11 e 12 asseguram que a jornada do motorista, o tempo máximo do usuário dentro do veículo, a capacidade do veículo e as janelas de tempo sejam respeitadas.

2.4. Heurística Construtiva e Busca Local

A heurística construtiva é um tipo de algoritmo construtivo que geralmente parte de uma solução vazia e constrói a rota, inserindo sempre uma solicitação de cada vez baseado em alguns critérios específicos do problema que esteja sendo resolvido até esgotar as possibilidades ou atingir um critério de parada. Este método permite que um algoritmo possa começar com uma

solução inicial factível e que outros algoritmos de melhoramento possam trabalhar com a solução produzida.

Métodos de busca local são heurísticas de melhoramento que realizam basicamente alterações em uma solução já existente dentro do alcance de sua vizinhança $N(x_L)$. Toda nova solução gerada x' a partir de algum procedimento de alteração da solução de referência da busca local x_L é uma solução vizinha. Assim, a vizinhança se constitui de todas as possíveis novas soluções resultantes das modificações possíveis na solução que a busca local está apontando como referência. Uma solução ótima local é a melhor solução dentro do conjunto da vizinhança. A solução só poderá ser dita ótima global se esta for a melhor solução dentro do conjunto de todas as possíveis soluções do problema e não somente da vizinhança. Ao longo do processo as melhores soluções vão sendo armazenadas e a vizinhança vai mudando na busca por melhores soluções.

3. Solução Desenvolvida

O desenvolvimento da ferramenta computacional para auxiliar o Programa de Acessibilidade Especial porta a porta – PRAE do município de Natal/RN fez parte de um projeto de extensão da Universidade Federal do Rio Grande do Norte criado para melhorar o desempenho do programa. Este projeto teve como objetivo desenvolver ferramentas computacionais de apoio à decisão para o PRAE, um serviço de transporte gratuito, oferecido pela Prefeitura Municipal do Natal à população com necessidades especiais de locomoção. O agendamento do serviço é realizado via telefone, podendo se dar em duas formas: (i) agendamento de uso fixo/periódico (e.g. sessões de quimioterapia, hemodiálise), e (ii) agendamentos eventuais. Em virtude de sua alta complexidade, o planejamento logístico do PRAE demanda o uso de técnicas de Pesquisa Operacional com o intuito de proporcionar maior produtividade e dinamismo à esta etapa decisória do programa. Articulado atividades de ensino, pesquisa e extensão de forma indissociável, este projeto permitirá alcançar boas soluções do ponto de vista operacional e logístico, de modo a aumentar a abrangência e democratização deste serviço de transporte municipal urbano.

Para armazenar as informações das instâncias foi criado um banco de dados com a base SQLite e um sistema de interface com o usuário via *Browser*, o qual foi programado na linguagem de programação *Ruby on Rails*. A principal função deste sistema é coletar informações sobre a distância entre todos os usuários, hospitais e escolas. Para isso, o sistema se utiliza da versão gratuita da aplicação online *Google Maps*.

Este sistema fornece ao *Google Maps* informações como endereço, Código Postal, e posição geográfica aos pares e coleta as informações sobre a distância entre eles. Em seguida esta informação é armazenada numa matriz $m \times m$ chamada de **Matriz de Distâncias**. Esta matriz é a representação computacional do grafo do problema, onde m são todos os lugares envolvidos na programação, inclusive as garagens. A matriz precisa ser completa porque o grafo é direcional, ou seja, ocorre de, entre duas localidades, o caminho de ida ter uma distância diferente do caminho de volta devido o sentido obrigatório de algumas estradas. Vale ressaltar que as distâncias coletadas podem ser consideradas como o caminho mais curto entre os dois pontos, entretanto, este processo não prevê fatores como trânsito, ou mudanças recentes na direção das estradas. Esta limitação, contudo, não invalidou o modelo de distâncias adotado pois os gestores do PRAE também se utilizam da mesma fonte de informação, e variações recentes ou temporárias nos trajetos são incorporadas nas medidas corretivas da programação final.

O algoritmo desenvolvido é composto de dois procedimentos principais: a heurística construtiva, e a busca local. A princípio, foi tomada como base de referência o método utilizado por Parragh *et al.* (2010). Entretanto, durante a concepção do modelo foi possível perceber que, assim como outros problemas do tipo *dial-a-ride*, o problema do PRAE exigiria adaptações para seu caso específico.

As principais mudanças foram relacionadas à função objetivo e à representação da rota. Na literatura referência, o modelo foi concebido para resolver a situação na qual um conjunto de solicitações são todas atendidas procurando minimizar os custos daquele dia onde a cada dia o algoritmo é utilizado novamente para definir as rotas e, neste caso, as solicitações são pontuais e não se repetem ao longo do tempo, por exemplo, os casos de emergência. A programação do serviço de transporte do PRAE não inclui transporte emergencial, os passageiros possuem uma agenda definida e foram contemplados por apresentarem mobilidade reduzida e impossibilidade de se utilizar do transporte convencional. Assim, a programação é estabelecida para uma semana inteira, e qualquer alteração que se faça necessária é implementada apenas na semana seguinte, caracterizando a versão estática do DARP. Além disso, o tratamento de um paciente no caso do PRAE se dá em mais de uma semana podendo se estender por meses onde, por razões de ordem jurídica, os pedidos em fase de atendimento pelo PRAE não podem ser removidos sob pena de intervenção do ministério público configurando, assim, solicitações fixas enrijecendo o número de combinações disponíveis para o algoritmo trabalhar. Quando há a inclusão de um usuário no agendamento semanal, são os assistentes sociais que decidem quem

será contemplado, ou seja, quem passará a ser atendido pelo serviço, a partir de uma lista de espera que fica na sede em pastas distribuídas por regiões do município de Natal. A lista de espera funciona como uma fila, entretanto a posição das pessoas que se cadastram diariamente nesta lista pode variar a critério de decisão dos assistentes sociais em função do grau de necessidade do indivíduo. Os motoristas recebem do escritório uma tabela com a programação, chamada de roteiro, para trabalhar e seguir com ela. Estas tabelas são atualizadas semanalmente ou quando há a necessidade.

Diante do exposto acima, a função objetivo do algoritmo desenvolvido procura a maximização dos pedidos atendidos dada uma frota limitada e pedidos em espera classificados em uma fila de acordo com a prioridade de atendimento definida pelos gestores do PRAE. Nem todas as solicitações poderão ser atendidas e a importância maior passa a ser o número de pessoas que possam ser contempladas e não mais os custos do serviço. No modelo clássico todos os pedidos são direcionados para um dos veículos para aquele único dia da semana e no dia seguinte o processo se repete. No PRAE um pedido é direcionado para mais de um dia da semana e a programação é semanal.

O modelo desenvolvido nesta pesquisa procura trabalhar a programação dos roteiros da mesma forma que é feita no problema real. Assim, uma solução gerada pelo algoritmo também resultará em uma programação para toda a semana. Desta forma, a variável de decisão precisa de índices que represente um veículo específico para um dos turnos do dia, e para um dia da semana em particular, de modo a cobrir todos os dias úteis da semana. Este aspecto da programação para uma semana inteira é sem paralelo na literatura sobre o DARP e foi justamente a adaptação que mais consumiu tempo e recursos para ser desenvolvida.

Além disso, um usuário pode ter mais de uma solicitação, por exemplo, a mesma pessoa pode ter uma requisição de transporte para o turno da manhã nas segundas e quartas-feiras para um local A, e ter outra solicitação de transporte às sextas-feiras para um local B. Além disso, os pedidos podem ser para o transporte só de ida (*Outbound*), ou só de volta (*Inbound*), ou ambos. Quando um usuário possui uma solicitação de ida e volta, o local de origem é sempre o mesmo do de retorno. Porém, é possível que a solicitação de ida esteja alocada em um veículo diferente da solicitação de volta. Logo, como cada veículo é alocado a uma rota que por sua vez representa uma programação semanal, as solicitações do usuário podem ser alocadas em rotas diferentes.

O exemplo reduzido a seguir mostra o funcionamento resumido da programação semanal. Supondo um serviço com apenas um veículo disponível das 7 às 8 horas da manhã

PESQUISA OPERACIONAL PARA O DESENVOLVIMENTO

para um único hospital. Apenas os pedidos P1 e P2 estão sendo atendidos atualmente. O primeiro quadro da figura 1 mostra as solicitações de cada pedido. A ordem em que os pedidos aparecem no quadro é o resultado da organização por prioridade definido pelos assistentes sociais. Existem mais três pedidos na fila de espera: P3, P4 e P5. Atualmente P1 precisa chegar ao seu destino às 8 horas e P2 às 8:15. O quadro superior direito mostra a programação de embarque (residência) e desembarque (hospital) para P1 e P2 e não podem ser removidos. Cada célula do quadro representa um intervalo de 15 minutos, assim como a janela de tempo (tolerância de atraso). A cor preenchida significa embarque ou desembarque do pedido ocorrido dentro daquele intervalo incluindo as extremidades. O algoritmo então tenta encaixar mais um pedido da fila na programação e consegue incluir P3 empurrando os horários de P1 e P2 dentro da janela de tempo permitida. Depois o algoritmo aloca P4 sem problemas. Por último o algoritmo tenta alocar P5, entretanto, só consegue alocar dois dias da semana sem violar as restrições enquanto que o pedido possui solicitações para todos os dias. Neste último caso, o algoritmo remove o pedido P5 por não haver possibilidade de atendê-lo nas condições desta combinação. Caso houvesse mais pedidos em fila outras combinações seriam tentadas.

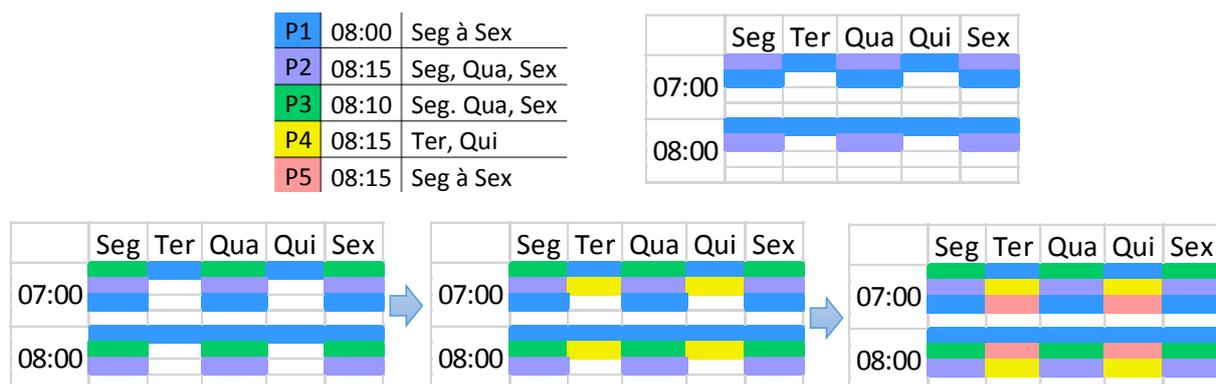


Figura 1: Programação semanal

O objetivo do algoritmo é maximizar o número de pedidos atendidos. Um pedido pode gerar várias solicitações, e são elas que são manipuladas pelo algoritmo. Por exemplo, um pedido de somente ida para um hospital na segunda-feira e terça-feira demanda duas solicitações de *outbound*. Outro pedido de ida e volta para apenas um dia da semana também gera duas solicitações, mas neste caso uma de *outbound* e outra de *inbound*. E ainda, um pedido de ida e volta tanto para o turno da manhã quanto para o turno da tarde gera quatro solicitações, duas de *outbound* e duas de *inbound*. Um pedido somente é considerado atendido quando todas as suas solicitações foram encaixadas com sucesso na programação dos veículos sem violar as

restrições impostas.

A premissa básica do algoritmo é manter agendados todos os usuários que já se utilizam do serviço, e ajustar as sequências de viagens das rotas procurando por brechas para encaixar mais solicitações atendendo a todas as restrições. Uma **Função de Avaliação** foi criada para calcular o número de pedidos atendidos pela solução corrente e penalizar as violações nas restrições do problema, assim como fez Cordeau e Laporte (2003b). Esta função se encontra detalhada no tópico 3.1.

Utiliza-se, também, o conceito de **vértice crítico** adotado por Cordeau e Laporte (2003b), onde são atribuídas janelas de tempo estreitas para os vértices de desembarque em um pedido de *outbound*, como também para os vértices de embarque em um pedido de *inbound*, ou seja, a janela de tempo é estreita para os locais de atendimento como hospitais e escolas. Entende-se que fora de sua residência, os usuários se sentem menos confortáveis para ficar esperando ser atendido ou para embarcar no veículo, portanto, a espera pelo veículo fora de sua residência deve ser reduzida.

3.1. Função de Avaliação

A função de avaliação é o termômetro do modelo e sempre que é executada pelo algoritmo são computados todos os tempos e violações das restrições:

$$f(s) = n^{\circ} \text{ de pedidos} - \alpha q(s) - \beta d(s) - \gamma w(s) - \tau t(s) \quad (1)$$

Os parâmetros, α , β , γ , e τ , são fatores de ponderação que penalizam, em diferentes graus, as quatro violações da equação. Estes valores são fixos.

Teste preliminares computacionais indicaram que os melhores valores para os parâmetros são: $\alpha = 1$; $\beta = 0,1$; $\gamma = 0,02$ e $\tau = 1$. Estes valores são constantes e indicam a força da restrição na função de avaliação. Para chegar nestes valores foram feitos testes limitados ao problema específico do PRAE. A medida que as soluções preliminares foram sendo apresentadas aos gestores do PRAE, os mesmos apontaram a importância de cada uma daquelas restrições de modo que se chegou ao consenso dos valores apresentados acima.

Para calcular cada uma das violações é preciso armazenar diferentes medidas de tempo atribuídas a cada um dos vértices da programação gerada pela solução. Estes tempos são calculados da mesma forma que no modelo clássico do DARF. Vale lembrar que t_{ij} é o tempo de deslocamento retirado da matriz das distâncias

Além dos tempos apresentados acima, o modelo também armazena a informação sobre o número de pessoas que estão dentro do veículo na partida do veículo de todos os vértices. O

número de pessoas dentro do veículo na saída de cada vértice é representado por y_i .

O conjunto de todos os vértices é representado por $\{0, 2n+1\}$, onde os intervalos; $[1, n]$ representa os vértices de embarque com carga positiva $+q$, e $[n, 2n]$ representa os vértices de desembarque com carga negativa $-q$. Uma solicitação é representada pelo par $(i, n + 1)$, e as garagens são representadas pelos vértices 0 e $2n + 1$. Neste problema, o número de rotas é representado por m . Assim, a carga no vértice i é calculada como sendo, $y_i = y_{i-1} \pm q_i$ para cada rota. As violações são, assim, calculadas:

Violação da capacidade do veículo é a soma dos valores em todos os momentos onde a carga máxima do veículo foi superada:

$$q(s) = \sum_{i=1}^{2n} (y_i - Q)^+ \quad (2)$$

Violação de janela de tempo é a soma dos valores em todos os momentos em que o início do serviço foi superior ao tempo mais tarde da janela de tempo:

$$w(s) = \sum_{i=1}^{2n} (B_i - l_i)^+ \quad (3)$$

Violação do tempo em que o paciente permanece dentro do veículo é a soma do tempo de todos os usuários que ultrapassaram o limite estabelecido:

$$t(s) = \sum_{i=1}^n (L_i - L)^+ \quad (4)$$

Violação do tempo de viagem total da rota é a soma dos tempos das rotas que ultrapassaram a carga horária do motorista:

$$d(s) = \sum_{k=1}^m (B_{2n+1}^k - B_0^k - T)^+ \quad (5)$$

3.2. Estratégia de Inserção

A estratégia de inserção é um procedimento usado tanto na Heurística Construtiva quanto na Busca Local e, sua finalidade é procurar, dentro da rota, todas as possíveis posições nas quais a capacidade máxima do veículo não é ultrapassada. Esta função recebe como parâmetro de entrada um par de vértices que compõe uma solicitação $(i, n + 1)$, com um vértice crítico e seu correspondente não crítico.

As solicitações novas sempre serão fornecidas por uma lista de espera que representa os usuários que ainda não foram contemplados pelo serviço do PRAE. A organização desta lista é tratada no tópico 3.3.

O procedimento de inserção se utiliza das informações armazenadas sobre a carga atual

do veículo y_i apresentada no tópico 3.1.

O algoritmo identifica as posições para as quais o par de vértices pode ser inserido sem violar a restrição de carga procurando pelos momentos em que o veículo se encontra lotado, ou que ao adicionar a nova carga esses pontos ultrapassem a carga máxima permitida. Após isto, a função retorna as informações para os procedimentos que requisitaram este procedimento.

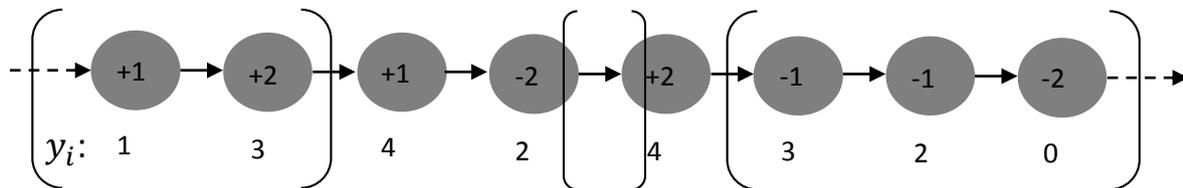


Figura 2: Funcionamento da estratégia de inserção

Por exemplo, na Figura 2, têm-se uma rota com a sequência da programação do roteiro e seus vértices de embarque e desembarque para um veículo com capacidade para 4 pessoas. Os valores de y_i mostram a quantidade de pessoas dentro do veículo, e em dois momentos o veículo se encontra lotado. Se um novo pedido de carga 1 for ser adicionado nesta rota, o par de vértices desta requisição só poderá ser inserido dentro de um dos espaços destacados em parênteses. Em outras palavras, a estratégia de inserção reduz o esforço do algoritmo direcionando a análise de combinações de possibilidade para as regiões destacadas porque a inserção nos trechos fora delas violariam a capacidade.

3.3. Heurística Construtiva

Este procedimento procura alocar solicitações que se encontram em uma lista de espera mediante a estratégia de inserção descrita no tópico anterior. A organização da fila de espera é feita considerando quatro critérios de ordenação na sequência de prioridade a seguir: 1º gravidade – grau da gravidade ou da urgência de atendimento de algum paciente, com valores entre 1 e 5; 2º status – é explicado mais abaixo, a prioridade é $1 < 0 < -1$; 3º ganha prioridade a solicitação com maior valor no cálculo: quantidade de dias que conseguiu encaixar / quantidade de dias requisitados; 4º a data de solicitação mais distante da data atual.

Um dos critérios de ordenação, que foi chamado de Status, foi criado no intuito de priorizar solicitações que já foram alocadas em algum dia da semana. O critério é composto de 3 campos: Status (mesmo nome do critério pai); Quantidade de Solicitações; e Dias.

O campo status possui três estados: 0 : a solicitação ainda não foi tentada e aguarda na lista; 1: a solicitação já foi tentada e até o presente dia da semana conseguiu ser encaixada nos

dias requisitados e; -1: a solicitação já foi tentada e não foi capaz de encaixar em algum horário requisitado. O campo Quantidade de solicitações indica se o pedido é somente de ida (direcional), ou de ida e volta (bidirecional). O campo Dias armazena a informação de quantos dias da semana o pedido conseguiu encaixar suas solicitações. Este critério funciona da seguinte maneira: 1º Todas as solicitações da fila iniciam com status 0; 2º Quando uma solicitação é selecionada e conseguiu ser encaixada, ela muda para status 1; 3º Se em qualquer uma das tentativas de alocar a solicitação na rota falhou ela recebe status -1; 4º Se na seleção de uma requisição ela tiver status -1, ela é imediatamente descartada, e o algoritmo passa para a próxima na fila. A figura 3 mostra o critério esquematizado.

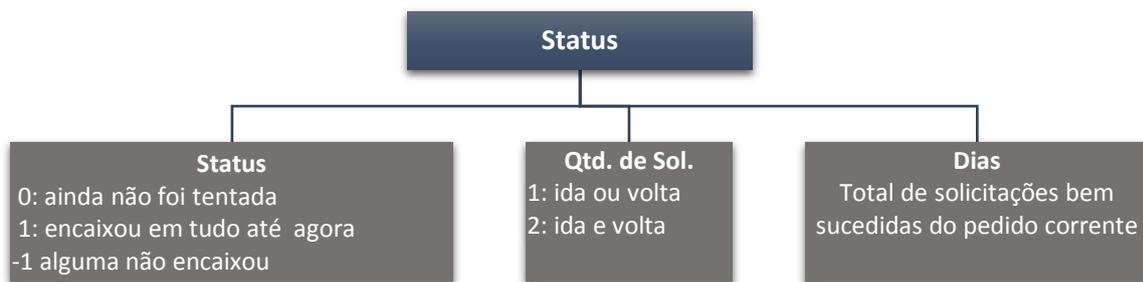


Figura 3 – Funcionamento do critério status

Na medida em que os valores do status vão sendo alterados, a lista de espera vai se reordenando, priorizando aquelas com status 1.

A inserção de novos pedidos é feita em dois momentos distintos dentro do modelo. A inserção chamada neste trabalho de **HC** procura alocar novas requisições dentro das rotas respeitando todas as restrições impostas ao problema. Já o procedimento nomeado de **INSERÇÃO** permite que os tempos de permanência dos usuários dentro do veículo sejam violados, para em seguida ser tentado um processo de viabilização por meio da **Busca Local - BL**. É de conhecimento comum entre os pesquisadores que trabalham com heurísticas que permitir violações durante o processo de resolução contribui para que o modelo possa “escapar” de soluções ótimas locais.

O que a HC e a INSERÇÃO fazem em comum é justamente o processo de inserção de novas solicitações nas rotas para cada dia da semana. A programação das rotas do PRAE é feita baseada no planejamento da semana anterior, aplicando-se a ela as alterações devidas. Da mesma forma, a HC recebe como dados de entrada uma rota já preenchida com a programação do roteiro da semana anterior. Essa medida garante que a primeira solução seja factível já que foi utilizada anteriormente. Já a INSERÇÃO recebe a solução gerada pela HC.

PESQUISA OPERACIONAL PARA O DESENVOLVIMENTO

O procedimento sistemático da HC consiste em visitar, para cada dia da semana, todas as rotas, uma por vez, e em cada rota:

1. Elege uma solicitação da lista de espera;
2. Lista as possíveis posições em que a requisição pode ser inserida por meio da estratégia de inserção;
3. Insere o vértice crítico na primeira posição possível e seu respectivo não crítico;
4. Verifica o valor da função de avaliação;
5. Se a função não for viável as próximas posições são tentadas;
6. Se as tentativas se esgotaram e não foi encontrada solução viável, esta solicitação é descartada e, repetem-se os passos de 1 a 5;
7. Quando uma solução viável é encontrada o pedido é mantido na posição, e repetem-se os passos de 1 a 6;
8. Quando todas as possibilidades da lista se esgotam, o algoritmo salta para a próxima rota e repete os passos anteriores.

A função INSERÇÃO é mais enxuta que a HC com as seguintes etapas:

1. Recebe uma solicitação;
2. Lista as possíveis posições em que a requisição pode ser inserida por meio da estratégia de inserção;
3. Insere o vértice crítico na primeira posição possível e seu respectivo não crítico;
4. Armazena o valor da função de avaliação desconsiderando violações no tempo de viagem;
5. Remove o par de vértices e tenta a próxima combinação possível repetindo os passos 3 e 4 até que não haja mais combinações possíveis;
6. Encaixa a solicitação na posição com o melhor valor da função de avaliação armazenada.

Na INSERÇÃO as violações no tempo de viagem do passageiro não são consideradas violações no cálculo da função de avaliação. Além disso, ela permite que uma solução inviável seja gerada. É importante destacar esta diferença para entender o código principal no próximo tópico. Além disso, vale lembrar que a heurística trabalha por dia em todas as rotas e depois passa para o próximo dia da semana repetindo os processos.

3.4. Busca Local

A função da busca local, neste modelo, é principalmente melhorar o valor da função de

avaliação. Caso a solução recebida seja inviável, a Busca Local também opera no sentido de tentar viabilizá-la.

Esta função passa por todas as rotas, uma por vez. Além disso, uma rota possui 7 programações diferentes, uma para cada dia da semana. Assim, são vistas todas as rotas da segunda-feira, depois todas da terça-feira e assim por diante:

1. O primeiro vértice crítico encontrado é removido e seu respectivo não crítico (o par que forma uma solicitação), esse vértice ganha status de visitado;
2. São listadas todas as possíveis posições que o vértice crítico pode ser reinserido sem provocar violação de carga máxima do veículo de acordo com a estratégia de inserção detalhada no tópico 3.2;
3. Em cada posição possível, o vértice crítico é inserido, o vértice não crítico correspondente é, então, encaixado na posição mais próxima possível do vértice crítico, e o valor da função de avaliação é calculado. As posições possíveis sem violar carga também são vistas no procedimento de inserção para o vértice não crítico;
4. Se em algum momento a função de avaliação apresentou melhora, o melhor posicionamento encontrado é armazenado. Então, o próximo vértice crítico é selecionado juntamente com o seu respectivo não crítico;
5. O processo se repete até que todos os vértices críticos da rota tenham sido visitados, e finalmente é feito o mesmo processo para as rotas seguintes.

Vale ressaltar que a estratégia de tentar trazer os vértices não críticos para próximo de seu par primeiro, para só depois tentar as posições mais distantes promove principalmente reduções no tempo em que o usuário permanece dentro do veículo.

3.5. Pseudocódigo

A programação que já é feita sem auxílio computacional pelos gestores do PRAE é a primeira solução inicial utilizada pelo algoritmo pois se ela está sendo executada dentro das restrições do serviço significa que é uma solução factível. Outra razão para se utilizar uma programação já desenvolvida pelos gestores é o fato das solicitações que já estão sendo atendidas serem proibidas de serem removidas ou alteradas por ordem jurídica. As soluções iniciais das semanas seguintes serão as programações otimizadas geradas pelo algoritmo referentes a semana anterior. Isso garante que as soluções iniciais sejam sempre factíveis. As solicitações desta solução inicial são chamadas de pedidos prioritários, uma vez que estes pedidos não podem ser removidos do agendamento - um paciente não pode ter o seu atendimento suspenso pelo

PRAE sob risco de intervenção do ministério público. Vale ressaltar que a função de avaliação também é chamada dentro das funções HC, INSERÇÃO, e BL.

Os passos do algoritmo principal estão descritos a seguir:

1. A solução inicial recebe a solução gerada da semana anterior;
2. Aplica-se a função HC nesta solução no intuito de inserir novas solicitações, o resultado desta função é uma solução viável;
3. Para todos os pedidos na lista de espera:
 - Aplica-se a função INSERÇÃO na solução fornecida pela HC, o resultado desta solução pode ser inviável;
 - Aplica-se a Busca Local na solução recebida pela INSERÇÃO na tentativa de viabilizá-la;
 - Se a função de avaliação está viável, esta passa a ser a nova solução;
 - O processo se repete até que todas as solicitações da lista de espera tenham sido visitadas.

A Figura 4 apresenta o pseudocódigo do algoritmo.

```

1.  $S \leftarrow S_{\text{prioritários}}$ ;
2.  $S \leftarrow HC(S)$ ; {solução factível}
3. PARA  $\forall i \in N_{\text{fila}}$  FAÇA;
4.      $S' \leftarrow S$ ;
5.      $S' \leftarrow INSERÇÃO(S')$ ; {permite violação}
6.      $S' \leftarrow BL(S')$ ; {tenta viabilizar}
7.     SE ( $F_{\text{avaliação}}(S')$  é viável) ENTÃO
8.          $S \leftarrow S'$ ;
9.     FIMSE
10. FIMPARA
11. Remove pedidos incompletos;
12. RETORNA( $S$ );
    
```

Figura 4: Pseudocódigo do algoritmo

3.6. Resultados

Para validar o algoritmo foram simuladas situações reais empregando os dados do PRAE, representando os casos de solicitações dos mais variados possíveis que ocorrem no agendamento do programa de transporte. Na literatura sobre o tema deste artigo é comum denominar estas situações de instâncias de teste. Estas instâncias foram elaboradas junto com

os gestores do serviço de transporte e os responsáveis pelos agendamentos dos pedidos com o intuito de refletir a maior parte dos desafios encontrados. A instância é composta de pedidos previamente alocados nas rotas denominados **pedidos fixos**, que não podem ser movidos de suas rotas, e uma lista de **pedidos novos** em uma fila de espera.

As três instâncias criadas foram, então, resolvidas tanto pelo algoritmo como pela equipe responsável pelos agendamentos de forma manual (forma atual do trabalho), sob algumas regras: a) o tempo de jornada do motorista não pode ser ultrapassado; b) O usuário não pode chegar mais do que 15 minutos atrasado ou adiantado em seu local desejado; c) O usuário não pode esperar mais do que 15 minutos pelo veículo após o término de seu atendimento (conforme o horário cadastrado); d) A capacidade do veículo para passageiros cadeirantes e não-cadeirantes não pode ser excedida; e) Os pedidos dos usuários com acompanhante devem ocupar 2 vagas no veículo; f) O tempo de embarque/desembarque é de 3 minutos para não cadeirantes e de 5 minutos para cadeirantes (valores médios levantados estatisticamente); g) Todos os veículos são micro-ônibus com capacidade para 3 cadeirantes e 10 não cadeirantes; h) Cada instância possui 5 veículos para rodarem de segunda-feira à sexta-feira no turno da manhã e tarde.

Uma rota equivale a 1 veículo para 1 dia da semana para 1 turno, o que resulta em um total de 50 rotas por instância. Durante a construção das instâncias, a proporção de pedidos fixos e novos foram sendo alterados da seguinte forma: a segunda instância teve seus pedidos fixos reduzidos em 25% com base na quantidade de pedidos fixos da primeira instância, estes pedidos foram convertidos em novos pedidos. A terceira instância, por sua vez, teve uma redução de 25% na quantidade de pedidos fixos com base na segunda instância e, mais uma vez, a lista de pedidos novos cresceu, de maneira que em cada instância o total de pedidos (fixos e novos) se mantivesse constante. O intuito era investigar se isso melhoraria a solução final e qual função do código influencia mais no resultado.

O algoritmo foi executado em um Notebook com Sistema Operacional Windows 7 Starter, 32bits, Processador Intel® Atom™ CPU D425 @ 1,80GHz 1,79GHz com 2 GB de memória RAM. O tempo decorrido para a solução da primeira instância foi de 33,165 segundos, da segunda instância foi de 81,559 segundos e, a terceira instância levou 41,892 segundos. A resolução das instâncias na forma tradicional levaram em média cinco dias, pois cada simulação abordava todo um novo cenário.

Como os pedidos fixos não podem mudar de rota, o esforço de solução ficou, então, para tentar encaixar a maior quantidade de pedidos novos possíveis sem violar nenhuma

PESQUISA OPERACIONAL PARA O DESENVOLVIMENTO

restrição. O quadro 1 apresenta a solução das três instâncias pelo método convencional atual e pelo algoritmo.

Número de Pedidos Novos Atendidos				Número de Solicitações Atendidas			
Método	Instância 1	Instância 2	Instância 3	Método	Instância 1	Instância 2	Instância 3
Algoritmo	18	25	26	Algoritmo	70	95	128
Manual	12	24	21	Manual	40	69	107
Total	24	35	42	Total	104	135	207

Quadro 1 – Comparação das soluções

Em todas as instâncias o algoritmo conseguiu melhores resultados do que o método convencional. A diferença maior pode ser notada no número de solicitações atendidas, por exemplo, na instância 2, tem-se apenas 1 pedido a mais atendido pelo algoritmo em comparação com os gestores, no entanto, para essa mesma instância, os pedidos que foram alocados pelo algoritmo são muitos mais “pesados”, ou seja, possuíam muito mais solicitações e isso se refletiu no resultado do quadro da direita com uma diferença de 26 solicitações.

O algoritmo conseguiu atender, dos pedidos novos disponíveis, 75% na instância 1, 71% na instância 2, e 62% na instância 3. Enquanto que o método tradicional atendeu 50% na instância 1, 69% na instância 2, e 50% na instância 3. Das 104 solicitações da instância 1, o algoritmo atendeu 67% e os gestores 38%. Na instância 2, o algoritmo atendeu 70% e os gestores 51% das 135 solicitações disponíveis. Na 3ª instância, das 207 solicitações, 62% foram atendidas pelo algoritmo, e 52% pelo método manual.

Concernente à análise da contribuição das funções do algoritmo, foram feitas comparações sobre a quantidade de novos pedidos alocados em dois momentos do modelo computacional: após a função HC e após a função da Busca Local. Os resultados estão expostos no Quadro 2.

	Pedidos atendidos			Composição da instância	
	Após HC	Após BL	Contribuição	Pedidos Fixos	Pedidos Novos
Instância 1	12	18	50%	46	24
Instância 2	13	25	92%	35	35
Instância 3	13	27	108%	27	43

Quadro 2: Contribuição das funções do algoritmo

É fácil notar que a Busca Local foi vital na construção da solução final do algoritmo. Apesar do número de novos pedidos ter crescido, a função HC não melhorou a solução da instância 2 para a 3. Já a Busca Local se tornou cada vez mais importante à medida que mais

pedidos ficaram a disposição para serem manipulados chegando a mais que dobrar a quantidade de pedidos atendidos.

5. Considerações finais

Esta pesquisa procurou desenvolver um modelo computacional capaz de incrementar o número de pessoas atendidas pelo serviço de transporte do Programa de Acessibilidade Especial porta a porta da cidade do Natal. Para tanto, seguiu-se um método de pesquisa no qual a comparação dos resultados obtidos pelo modelo computacional desenvolvido e os resultados de um caso real serviram de instrumento para a medição das possíveis vantagens deste modelo. O algoritmo desenvolvido neste trabalho se baseou, principalmente, nos estudos de Parragh *et al.* (2010) e, Cordeau e Laporte (2003b)

De acordo com os resultados apresentados no tópico 3.6, e considerando que as instâncias utilizadas para proceder com as análises no processo de validação do algoritmo representam a realidade do funcionamento dos agendamentos do PRAE, conclui-se que o modelo computacional desenvolvido cumpre seu objetivo de contribuir positivamente para o número de pessoas que podem ser atendidas pelo serviço de transporte do PRAE, visto que o algoritmo conseguiu melhores resultados do que o processo convencional de agendamento das rotas dos veículos.

Outra consideração importante a respeito dos resultados obtidos pelo algoritmo é a constatação da ocorrência nula de violações da restrição de capacidade máxima dos veículos. Nenhuma rota violou capacidade dos veículos, e o número de vagas disponíveis para não cadeirantes se apresentou muito superior à demanda. A maior ocorrência de pessoas ao mesmo tempo dentro do veículo se deu em um trecho percorrido com seis passageiros, o que representa apenas 46,15% da capacidade total do veículo. Observa-se, com isso, que os veículos estão possivelmente circulando com capacidade ociosa e uma recomposição da frota atual poderia ser pensada como uma alternativa de redução de custos em longo prazo. A presente pesquisa não se aprofundou nesta análise de capacidade excedente, contudo, esta observação se apresenta como uma sugestão de análise gerencial na busca por alternativas mais eficientes do transporte estudado. Vale destacar que, tendo, a situação simulada, sido construída com o intuito de representar o funcionamento do real, a observação deste parágrafo pode também representar a realidade do PRAE.

Agradecimentos

Este trabalho recebeu financiamento do MEC/SESU através do Programa de Apoio à Extensão Universitária – PROEXT 2010. O primeiro autor agradece a CAPES pela bolsa que recebeu durante seu mestrado. Os autores agradecem também aos alunos Thiago Jerônimo, Rafael Franzon, Leandro Rochink e Thiago Correia, que fizeram parte do desenvolvimento do projeto “Sistema Computacional de Apoio às Decisões de Planejamento Logístico do Programa de Acessibilidade Especial – PRAE – Porta a Porta”.

Referências

- Aldaihani M., Dessouky M.M. (2003) Hybrid scheduling methods for paratransit operations. *Comput Ind Eng*, 45:75–96.
- Balas, E., Toth, P. (1985) Branch and bound methods. In *The Traveling Salesman Problem*, Editado por Lawler, E. L.; Lenstra, J. K.; Rinnooy Kan, A. H. G. and Shmoys, D. B. John Wiley & Sons Ltd, 361-401.
- Baugh J.W., Krishna G., Kakivaya R., Stone J.R. (1998) Intractability of the dial-a-ride problem and a multiobjective solution using simulated annealing. *Eng Optim*, 30:91–123.
- Bodin, L. D., Sexton, T. (1986), The multi-vehicle subscriber dial-a-ride problem. *TIMS Studies in Management Science*, 2, 73–86.
- Bodin, L. D., Golden, B. Assad, A.; Ball, M. (1983) Routing and scheduling of vehicles and crews: the state of the art. *Computers & Operational Research*, vol. 10, nr2.
- Censo. (2012) Censo 2010 do Instituto Brasileiro de Geografia e Estatística. Disponível em <http://www.ibge.gov.br/censo2010>. Acesso em 10 de Junho de 2012.
- Cordeau, J. F., Laporte, G. (2003a), The Dial-a-Ride Problem (DARP): Variants, modeling issues and algorithms. *4OR - Quarterly Journal of the Belgian, French and Italian Operations Research Societies*, 1: 89-101.
- Cordeau, J. F. Laporte, G. (2003b), A tabu search heuristic for the static multi-vehicle dial-a-ride problem. *Transportation Research: Part B*, v. 37, n. 6, p. 579-594.
- Cordeau, J. F. (2006) A branch-and-cut algorithm for the dial-a-ride problem. *Operations Research*, v. 54, n. 3, p. 573-586.
- Cordeau, J.-F., Laporte, G. (2007), The dial-a-ride problem: models and algorithms. *Annals of Operations Research*, 153, 29-46.
- Dantzig, G. B., Ramser, J. H. (1959) The Truck Dispatching Problem. *Management Science*, v.6, n.1, pp. 80-91.
- Jørgensen R.M., Larsen J., Bergvinsdottir K.B. (2007) Solving the dial-a-ride problem using genetic algorithms. *J Oper Res Soc*, 58:1321–31.
- Melachrinoudis, E., Ilhan, A. B., Min, H. (2007), A dial-a-ride problem for client transportation in a health-care organization. *Computers & Operations Research*, 34, 742–759.

Padberg, M. Rinaldi, G. A. (1991), Branch-and-Cut Algorithm for the Resolution of Large-Scale Symmetric Traveling Salesman Problems, SIAM J. Review, vol. 33, pp. 60-100 .

Parragh, Sophie N., Doerner, Karl F., Hartl, Richard F. (2010), Variable neighborhood search for the dial-a-ride problem. Computers & Operations Research, 37, 6, 1129-1138.

Prae. (2003) Prefeitura Municipal do Natal. Disponível em: <http://www.natal.rn.gov.br/sttu2/paginas/ctd-714.html>. Acesso em 02 de maio de 2013.

Saxe, J.B. (1980), Dynamic programming algorithms for recognizing small-bandwidth graphs in polynomial time. SIAM J. Algebraic and Discrete Methods, 1:363-369.

Toth P., Vigo D. (1997) Heuristic algorithms for the handicapped persons transportation problem. Transp Sci, 31:60–71.

Wazlawick, R. S. (2008) Metodologia de pesquisa para ciência da computação. Rio de Janeiro: Elsevier.